# BlockTorrent: A privacy-preserving data availability protocol for multiple stakeholder scenarios

Ambrose Hill, Shailesh Mishra, Ali Dorri, Volkan Dedeoglu, Raja Jurdak, and Salil S. Kanhere
{*ambrose.hill,volkan.dedeoglu*}*@data61.csiro.au,* {*ali.dorri,r.jurdak*}*@qut.edu.au*
*mshailesh0511@iitkgp.ac.in, salil.kanhere@unsw.edu.au*

*Abstract*—**As industries across the globe continue to digitize their processes, the need for a mechanism to share private data between multiple stakeholders is becoming increasingly apparent. However, sharing data poses challenges around privacy and accessibility, particularly in the event of disputes between stakeholders with a shared interest, such as a supply chain. Using supply chains as a case study, we present BlockTorrent to address these challenges and facilitate data sharing between supply chain participants. BlockTorrent allows participants to securely share their data in near real-time with other participants without the risk of information leakage or data manipulation.**

## 1. Introduction

Existing supply chain mechanisms for data sharing and accessibility heavily rely on participant compliance. Supply chains involve a group of organisations that are responsible for facilitating the transfer of goods and information from suppliers to customers. Recently, IoT sensor devices have been integrated with some supply chains allowing for the automatic tracking of items during their journey from supplier to the customer. Participants willing to share real-time data amongst themselves will facilitate faster trades, enjoy lower operational costs and have the ability to detect and competently rectify delays. However, data sharing among participants in supply chains poses privacy & accessability challenges.

One potential technology to facilitate this sharing is Blockchain. Blockchain supports multi-stakeholder applications, such as a supply chain, with data immutability, audibility and access control. Blockchain based supply chains have seen increased interest due to their improved scalability & traceability [1]–[5].

This paper proposes BlockTorrent, a novel blockchain-based data management protocol, that enhances data availability while protecting the privacy of the participants. BlockTorrent is an integration of BitTorrent, a peer-to-peer file-sharing protocol and Blockchain. To ensure data accessibility, BlockTorrent distributes data among a set of peers in the blockchain network. The data is split and distributed between multiple peers to ensure accessibility in the case that peers go offline or begin to act maliciously.

## 2. BlockTorrent: A Privacy-preserving Data Availability Protocol

BlockTorrent is a privacy-preserving protocol that ensures data availability during audits by combining blockchain and BitTorrent [6]. BitTorrent has been proven to be an effective file-sharing protocol in peer-to-peer networks, and it further improves BlockTorrent's transfer times over the traditional client-server architecture, allowing for near real-time data sharing.

First, we will introduce the network layers and their interactions within the system architecture. BlockTorrent relies on three key components, the main chain, overlay network and the private database, set up in parallel to reduce bottlenecks in network traffic. Then, we will outline the functionality that the proposed system offers and describe how it solves the privacy-preserving data availability challenges.

### 2.1. System Architecture

First, we define the key entities of our solution:

*Participant:* Any organisation that is a part of the supply chain employing BlockTorrent. This could be the supplier, transporter, retailer or authority representing the local or state governments.

*Admin Node:* A node, or group of nodes, for each participant that is responsible for accessing all layers of the network. These nodes monitor and maintain the data-sharing mechanism for each organisation. To avoid centralisation within a single organisation, the role of the admin node can be split up between multiple entities and/or devices.

*Auditor:* A unique participant that is responsible for auditing the supply chain data and is the entity that is responsible for dispute resolution between participants.

*User:* All other users interacting with the system such as a buyer.

As mentioned, there are three key components: the main chain, private database and the overlay network.

*Main Chain:* Serves as the interface between organisations and BlockTorrent. Each participant of the supply chain has access to the main chain and the ability to read and write transactions. Participants are added to the main chain when they join the supply chain consortium and use it as the source of truth in the network. The main chain facilitates the sharing of encrypted data between the parties.

*Private Database:* The private database is used for each individual member's private business data. Its purpose is to facilitate the use of the data while concealing it from the public network. This can be any type of database that can support data access by a node capable of performing BlockTorrent functions.

*Overlay Network:* The overlay network facilitates all non-main chain communication and storage. The data is distributed on the overlay network is first chunked to simulate the small packets that are used in the BitTorrent protocol.

978-0-7381-1420-0/21/$31.00 ©2021 IEEE

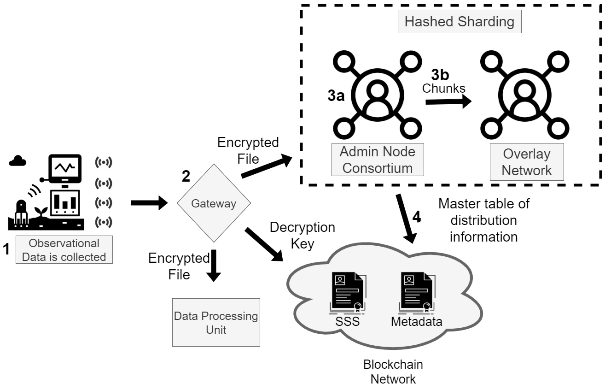

Figure 1: Transaction flow for storing data

## 2.2. Functionality

**2.2.1. Storing Files.** Fig. 1 shows the process a file undertakes during the storage process, including the collection, splitting and distribution of the file.
*Step 1*: New sensor data is collected and sent to the private database.
*Step 2*: The new data is detected by the admin node of the organisation. The data is first encrypted and then chunked. The encryption key is split into key share proofs using Shamir's sharing technique [7]. *Step 3a*: The admin node consortium while splitting the file and determining the owner peers, which are the peers that are designated to own a certain chunk. The admin also stores a record of each chunk's hash, owner peers and timestamp which is sent to the main chain. A master table is created for each file and updated with the record of each chunk. Each admin retains a copy of the master table until it is agreed upon and stored in the main chain. *Step 3b*: The admin peer then sends each chunk to the list of determined peers. *Step 4*: If this is the last chunk to be distributed then the master table is completed and submitted to the main chain. *Step 5*: The key share proofs are then submitted to the key management smart contract on the main chain that is responsible for sharing that key between participants.

**2.2.2. Recreating Files.** Fig.2 shows how an auditor can request both the key shares and chunks from other peers on the network, allowing access to the file without input from the original file owner.

*Step 1*: The admin requests the decryption key shares from the smart contract. Once requested, participants will be notified of the access request and if valid will submit a transaction saying their share of the key can be used, their share is included in the transaction. *Step 2*: Using the key, the admin retrieves the master table from the main chain, which contains chunk locations. *Step 3*: Combine the key share parts to create the complete decryption key. *Step 4*: For each chunk in the master table, the admin looks up its location, requests the chunk from the closest owner peer. *Step 5*: The auditor combines each chunk, computes the hash and compares it to the hash stored in the main chain. If they match, the auditor uses the key from Step 1 to decrypt the file. If any hash does not match its chunk, the auditor can request the chunk again until all chunks are acquired and validated.

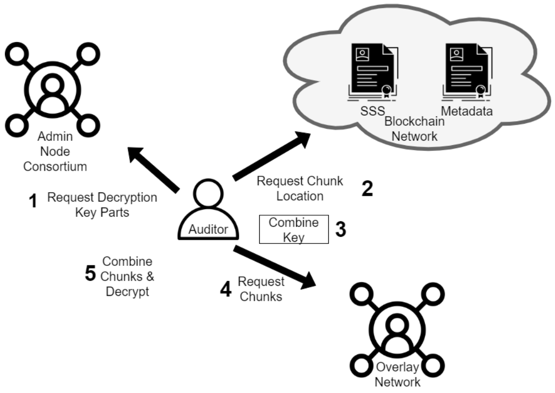

Figure 2: Transaction flow for storing and recreating data

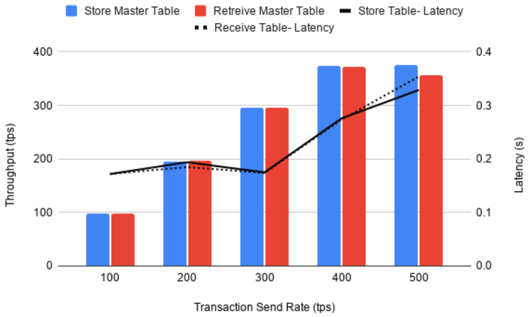

Figure 3: Transaction throughput and latency for the main blockchain layer

# 3. Implementation and Evaluation

## 3.1. Implementation

For our implementation, we assume each participant is managing their private database. The main chain was developed using Hyperledger Fabric (HLF) version 2.2 [8]. The smart contracts deployed on the main chain were written in go v1.11.2. The overlay network was written in Python 3.7. We made use of Python's native networking to simulate a peer-to-peer network as well as generate files to share. For encryption, we used the SHA3-256 algorithm.

## 3.2. Evaluation

Figure 3 shows the network performance results when tested with HLF's testing suite Caliper [9] and shows that Blocktorrent throughput peaks at around 350 tps and consistently has less than 0.4 s latency. Depending on how many participants are simultaneously using BlockTorrent, this throughput would be sufficient, as only the key share proofs and master table are being processed through the mainchain.

# 4. Conclusion

This poster presents BlockTorrent as a novel privacy-preserving data availability protocol. Rather than using the traditional client-server architecture BlockTorrent integrates BitTorrent and Blockchain to create multiple peer-to-peer networks that are leveraged to distribute and access private information promptly.

# References

[1] S. Malik, S. S. Kanhere, and R. Jurdak, “ProductChain: Scalable blockchain framework to support provenance in supply chains,” *NCA 2018 - 2018 IEEE 17th International Symposium on Network Computing and Applications*, 2018.

[2] K. Korpela, J. Hallikas, and T. Dahlberg, “Digital Supply Chain Transformation toward Blockchain Integration,” in *Proceedings of the 50th Hawaii International Conference on System Sciences (2017)*. Hawaii International Conference on System Sciences, 2017.

[3] R. Monfared and S. Abeyratne, “Blockchain ready manufacturing supply chain using distributed ledger,” *International Journal of Research in Engineering and Technology-IJRET*, no. 09, pp. 1–10, 2016. [Online]. Available: http://esatjournals.net/ijret/2016v05/i09/IJRET20160509001.pdf metadatarecord:https://dspace.lboro.ac.uk/2134/22625

[4] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, “Applications of Blockchains in the Internet of Things: A Comprehensive Survey,” pp. 1676–1717, 2019.

[5] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, “Trustchain: Trust management in blockchain and iot supported supply chains,” in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 184–193.

[6] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, “The Bittorrent P2P file-sharing system: Measurements and analysis,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3640 LNCS, pp. 205–216, 2005.

[7] A. Shamir, “How to Share a Secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 11 1979.

[8] IBM, “Hyperledger fabric blockchain platform,” 2020. [Online]. Available: https://www.hyperledger.org/projects/fabric

[9] Hyperledger, “Hyperledger caliper,” 2020. [Online]. Available: https://hyperledger.github.io/caliper/